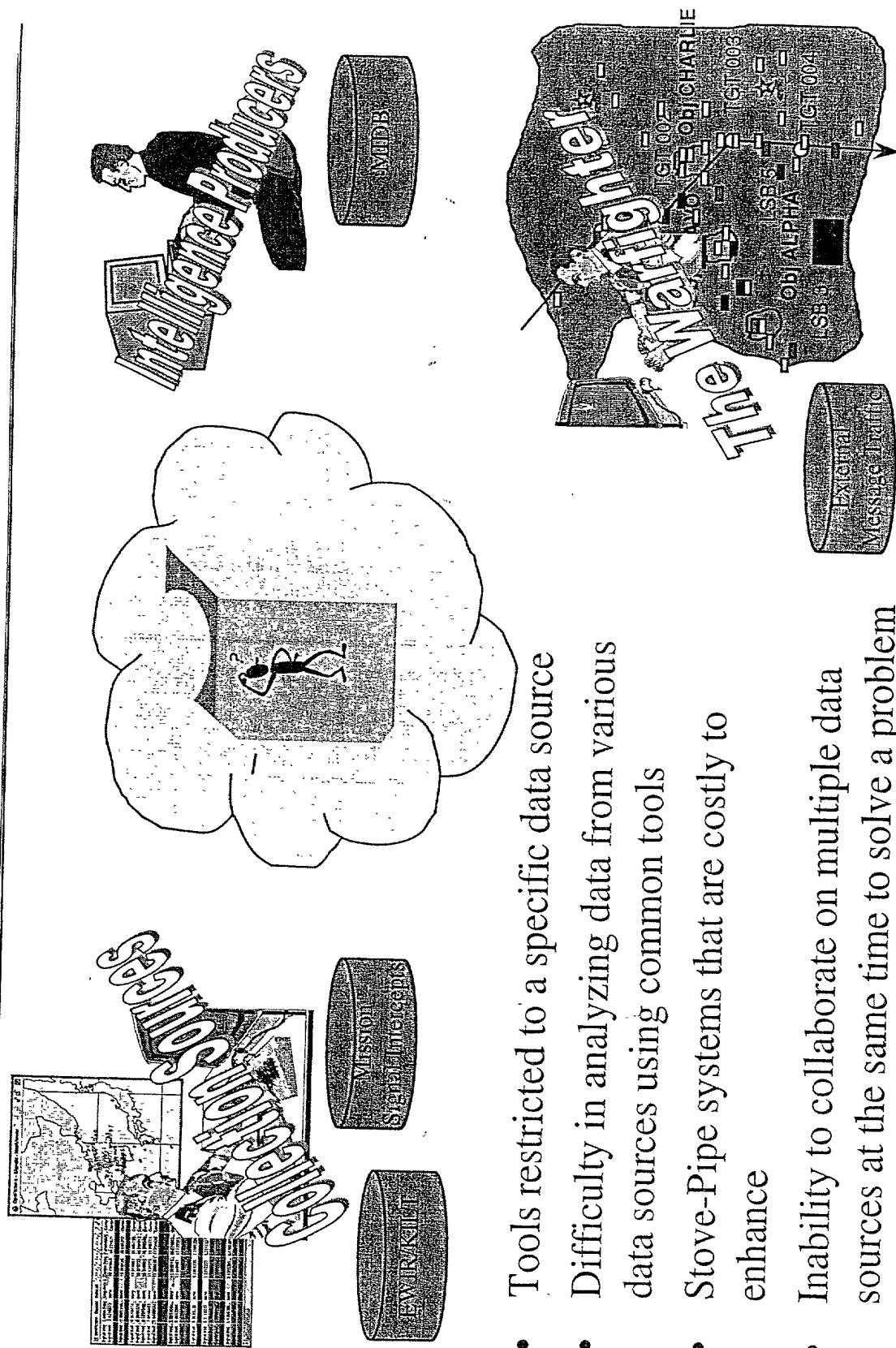TODAY

- Tools restricted to a specific data source
- Difficulty in analyzing data from various data sources using common tools
- Stove-Pipe systems that are costly to enhance
- Inability to collaborate on multiple data sources at the same time to solve a problem

Fig 1
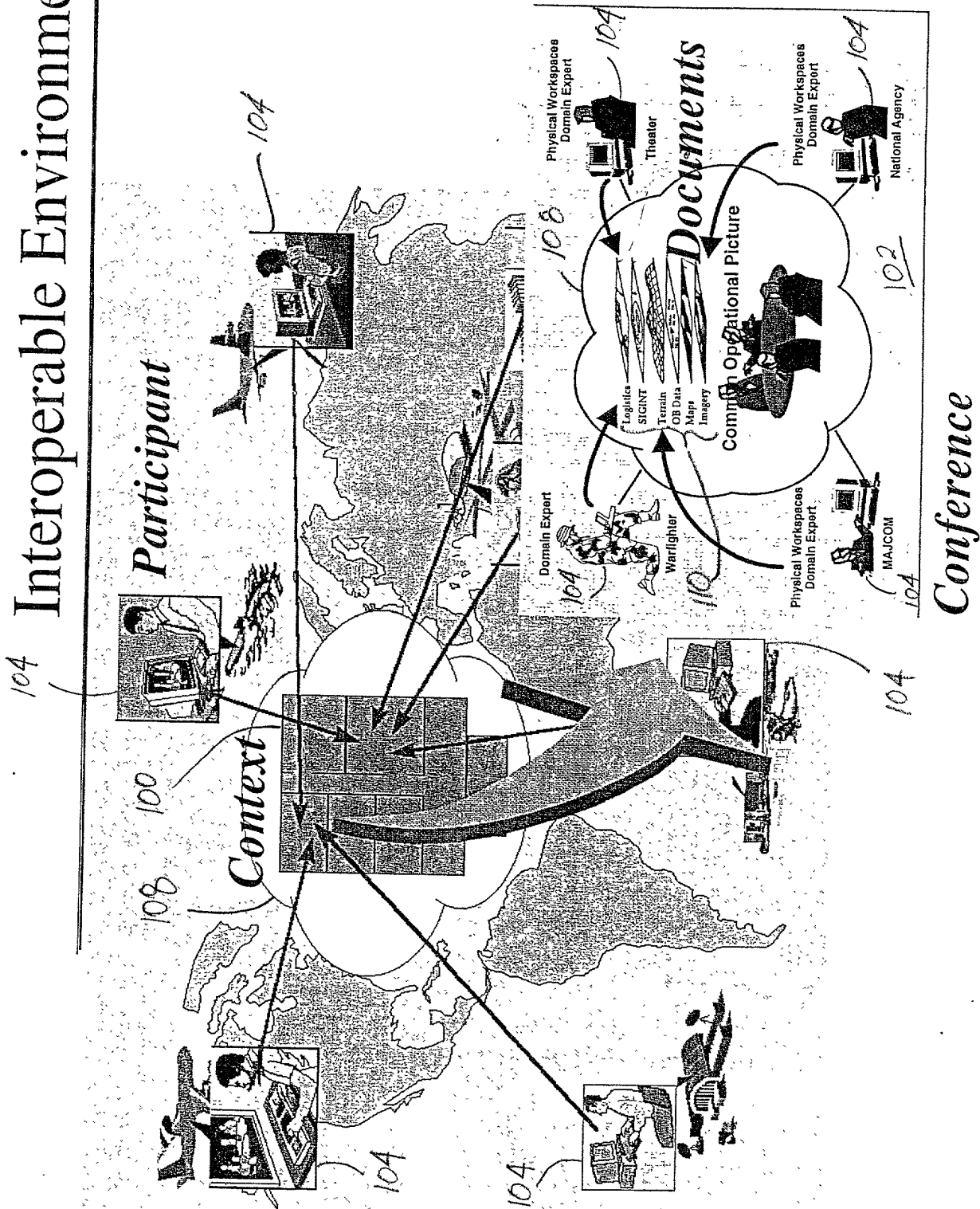
Collaborative Interoperable Environment

Participant

Context

Documents

Conference

Fig 2

"Document" based data manipulation

28

20

104

106

112

Document Management

10

Data Source

- A Document represents a collection of data

- Documents may be created by querying a data source

  − Standing queries are constantly evaluated

  − Static queries represent the state of the data source at the time the query was initiated

- Documents may initially be empty and populated through user or agent actions

- Documents are placed in conferences to be manipulated using the tools that best solve the problem.

Fig 3

Thin Clients interact with data represented by a document

10

18

Domain Data Is Extended For Visualization and Control
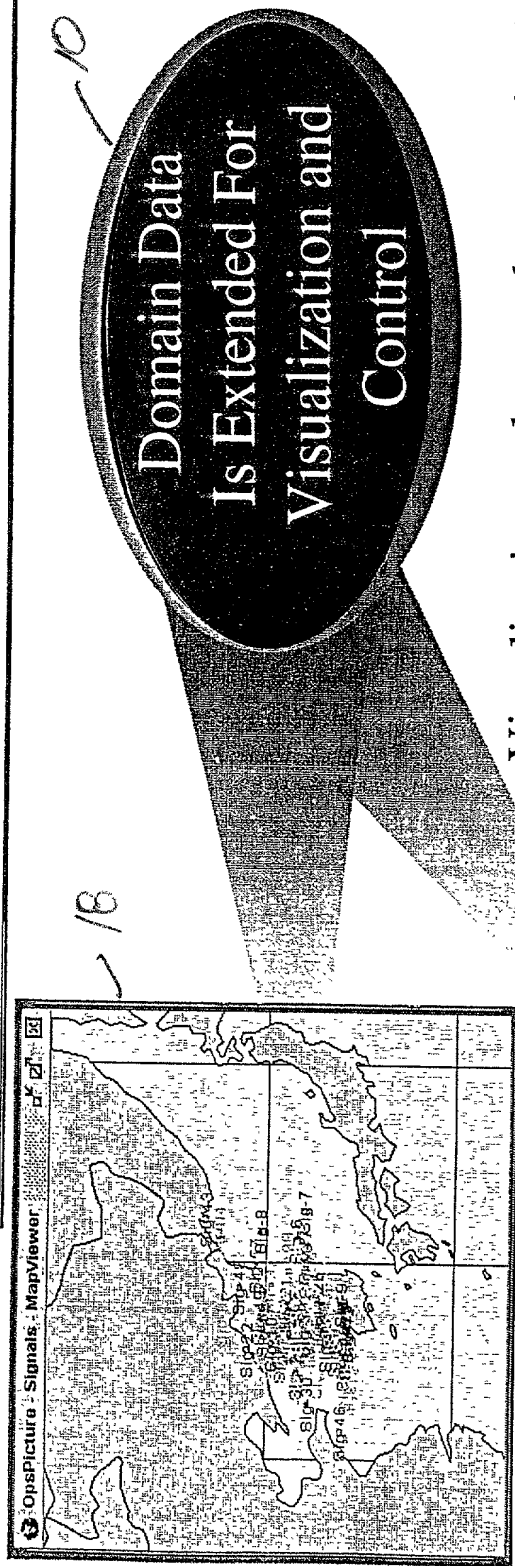
- Visualization and control properties (e.g., color, selection, symbol, etc.) become part of the data

- Client viewers focus on presentation of information

- Clients do not require logic dealing with collaboration

- Clients do not require complex logic to access data
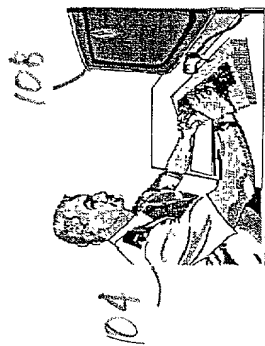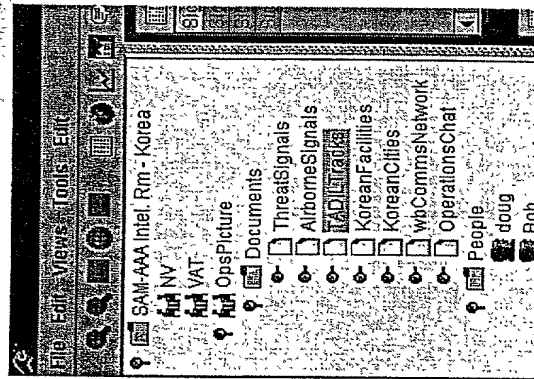
Displaying documents using various tools

OpsPicture : Signals : MapViewer

SAM-AAA Intel Rm - Korea
File Edit Views Tools Edit
NV
VAT
OpsPicture
Documents
ThreatSignals
AirborneSignals
Tracks
KoreanFacilities
KoreanCities
wbCommsNetwork
OperationsChat
People
doug
Bob

104

106

60

Fig 4

Collaboration on Multiple Views
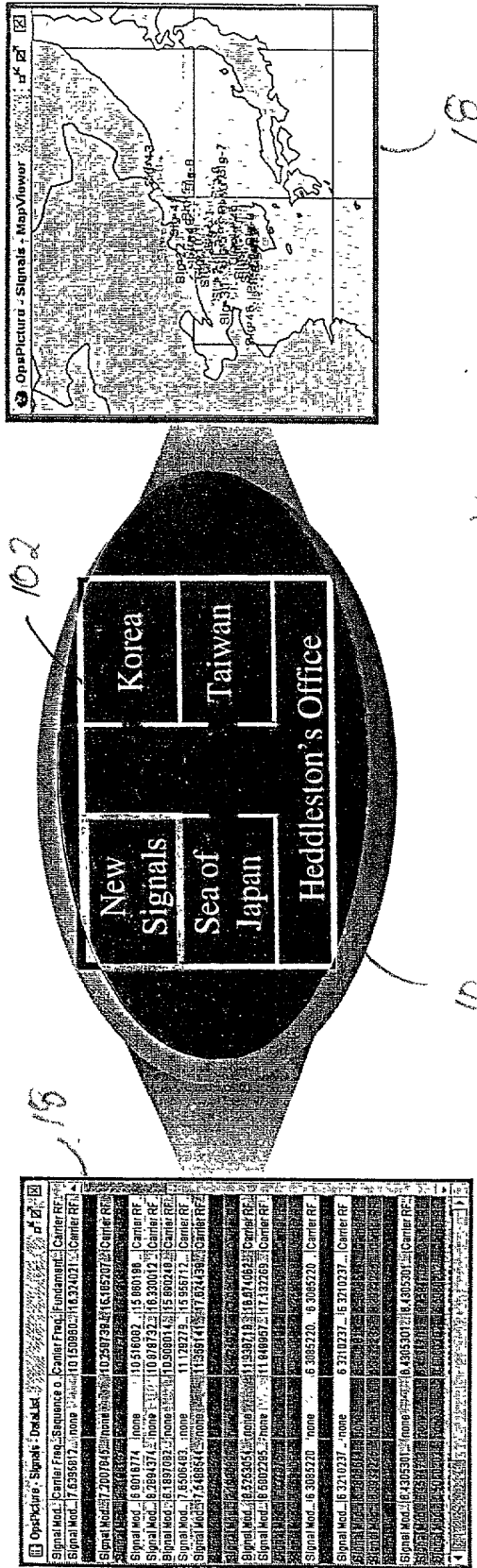
- Single user collaboration
- Multiple tools in the same conference coordinate visualization (e.g. highlight, color)
- All tools in a conference cooperate for problem solving
- No tool-to-tool communication

Fig. 5

Flexibility and Collaboration

Agents

ATO

Force Status

SAMs

Inci-dents

Northern NFZ
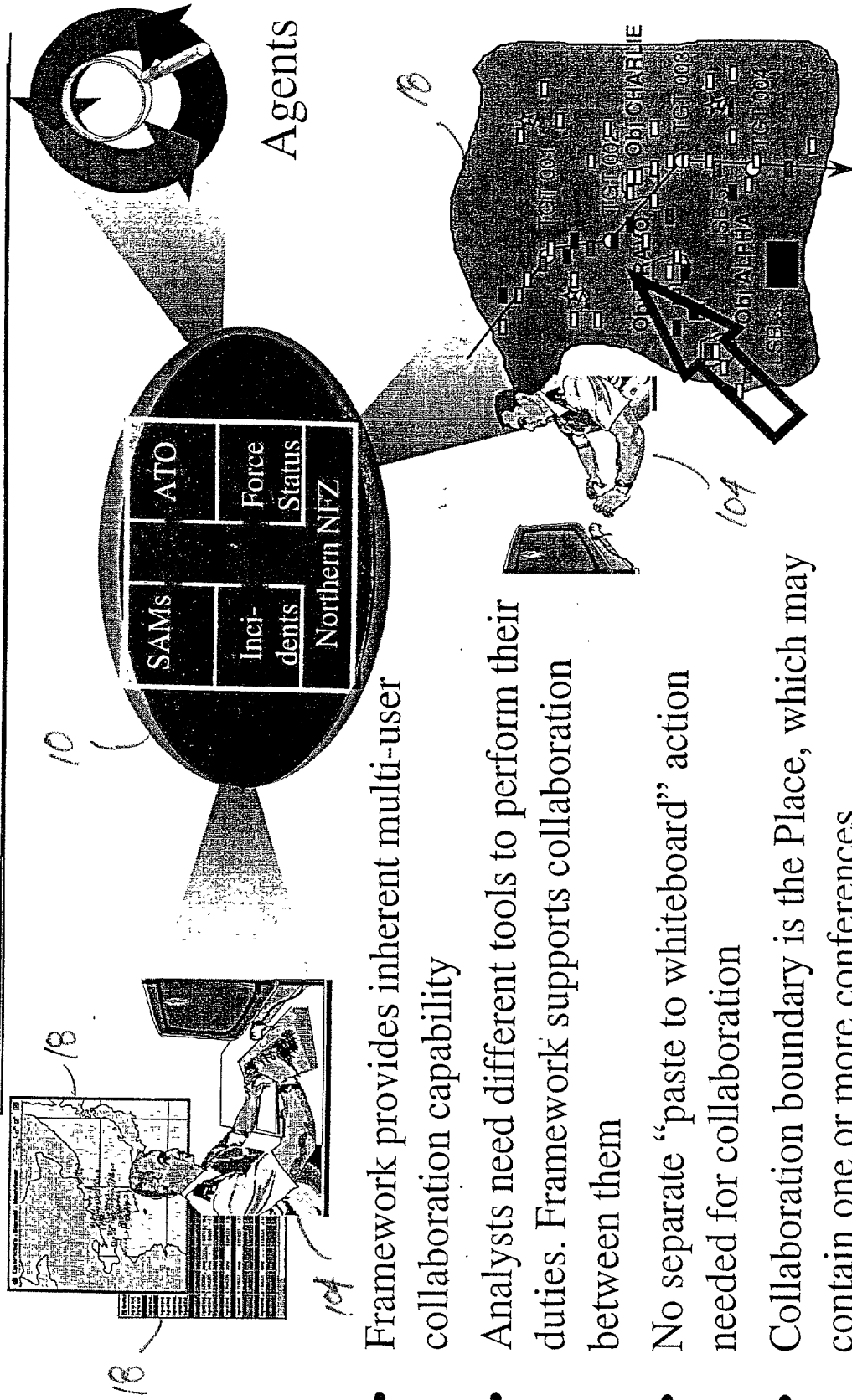
- Framework provides inherent multi-user collaboration capability

- Analysts need different tools to perform their duties. Framework supports collaboration between them

- No separate "paste to whiteboard" action needed for collaboration

- Collaboration boundary is the Place, which may contain one or more conferences

- Collaborators may be agents as well as humans

Fig 6

Collaboration Summary

Access To Intelligence Producers

Access To The Warfighter

Access To Assessors

Access To Information

Disarmament Room

Mine Operations Room

Shelter & Repatriation Room

Force Status Room

Kosovo Compliance Mission Floor

Fig. 7

Architectural Strategy

Key Reference Architectures
- Object Management Architecture (OMA)
  - OpenGIS, CosServices
- COE Layered Architecture
- UCA Cryptologic Framework
- USIGS
  - GIAS

Key Data Models
- SOM, MIDB, JCDB, ASAS, L245, ECDS, TEXTA

Architectural Patterns
- Layered Architecture
- Data Centric Architecture
  - Information Management Framework
  - Interactive Analysis Framework
- Mission Management Architecture
  - Task Management Framework
  - Resource Management Framework

COTS SW Infrastructure
- JAVA/C++
- CORBA
- Enterprise Java Beans
- RDBMS/ODBMS
- Microsoft Windows
- WEB Server/Browser
- XML / DOM

COTS HW
- UNIX SMP Server
- NT Workstations

Services Based Architecture

10

JCS Participant Server 24

JCS Context Server 26

JCS Folder Server 26

Signal Repository Server 12

Signal Files 112

Simple DB Repository Server 12

DB Files 112

Conference Manager 22

Map Viewer 16

Data Channel Server 14

List Viewer 18

Library Server 10

14

Query Viewer 20

System Service Activation Daemon 34

Factory Finder Service 32

Naming Service 30

108

16A

56	User Interface Tier
	User Interface
54	Services Tier
	Collaborative Management
	Channel Services
	Common System Services
52	Information Access Tier
	Information Access Services
	Data Management

Fig. 9

Extending The Infrastructure
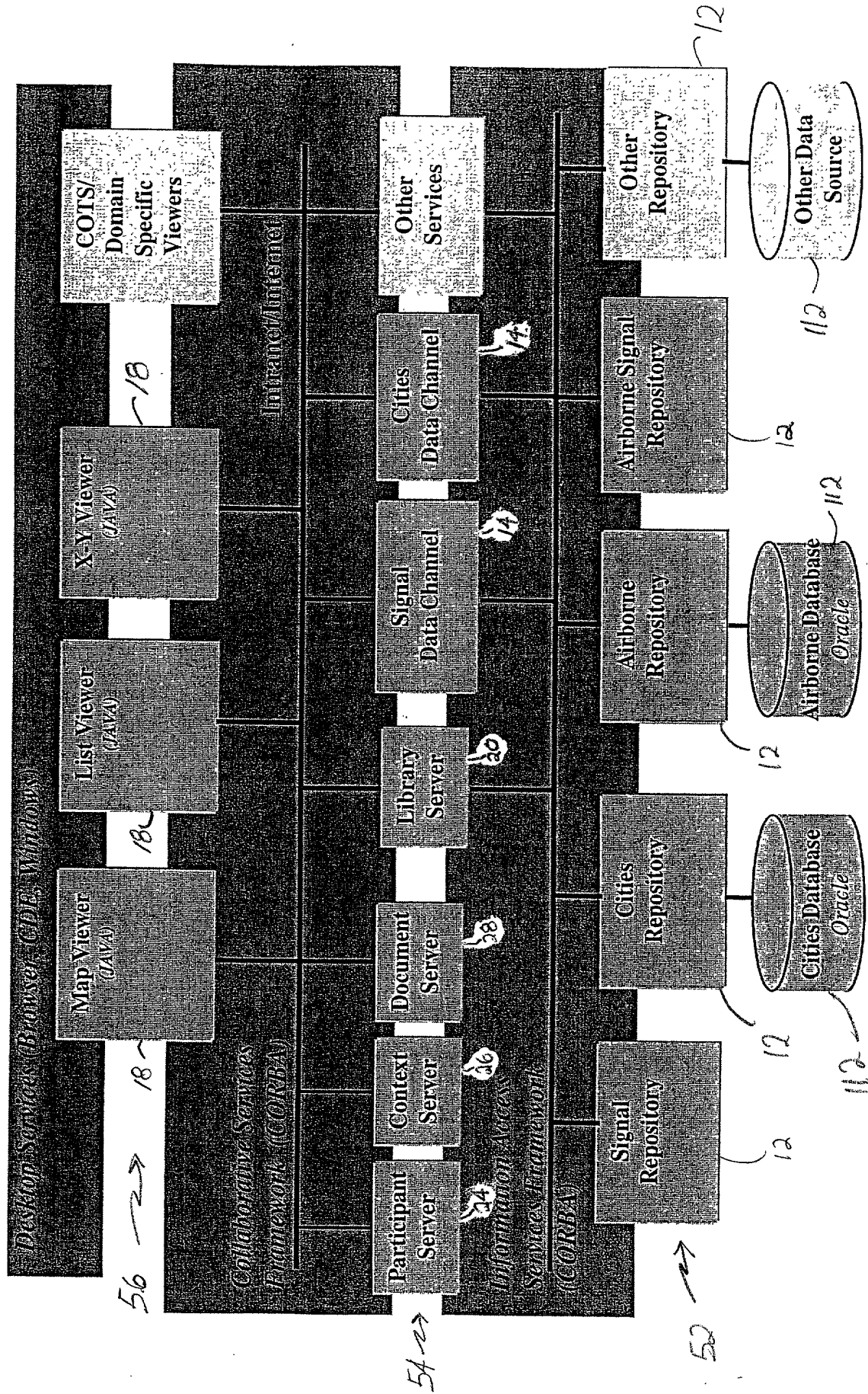
Fig. 10

Integration with legacy systems

Minimum Level Integration
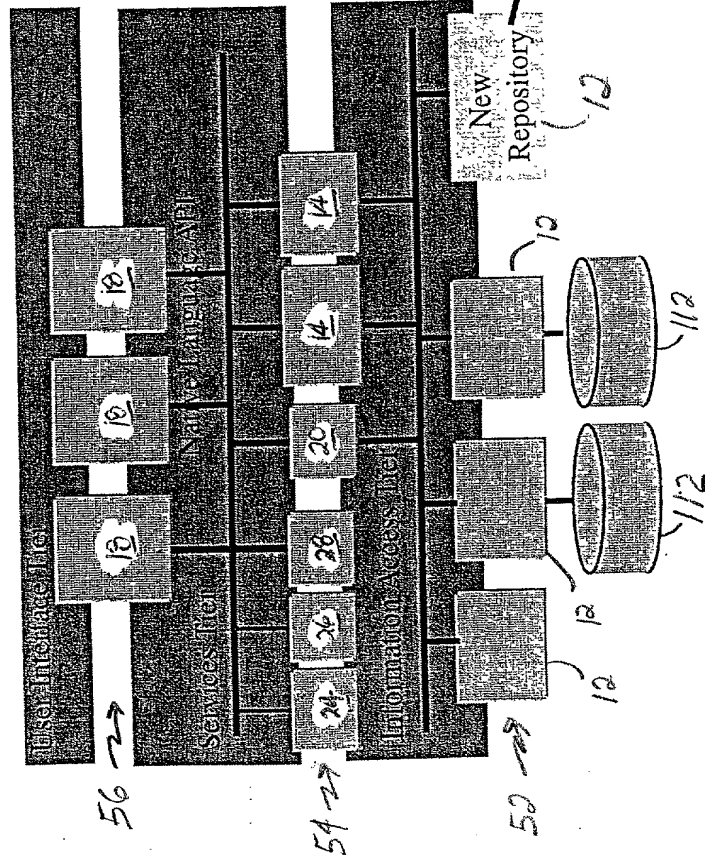
- Provide access to legacy data source through a new repository
- No legacy software changes required
- New data source is available for collaborative processing
- Provides new options for extending system capabilities
- Low/No Risk implementation

Legacy Application 202

Legacy Data Source 200

Legacy System

New Repository 12

User Interface Tier

New Language API

Services Tier

Information Access Tier

10

12

12

12

14

14

20

28

26

24

112

112

56

54

53

SYSTEM Infrastructure

Fig 11

Integration with legacy systems

Mid-Level Integration

- Access data through Tsunami infrastructure
- Legacy viewers are now interact collaboratively
- Still maintain the option to interact directly with the data source
- Provides additional options for extending system capabilities

Legacy Application

Legacy Data Source

New Repository

Native Java Bridge API

User Interface Tier

Services Tier

Information Access Tier

SYSTEM Infrastructure   Legacy System

Fig 12
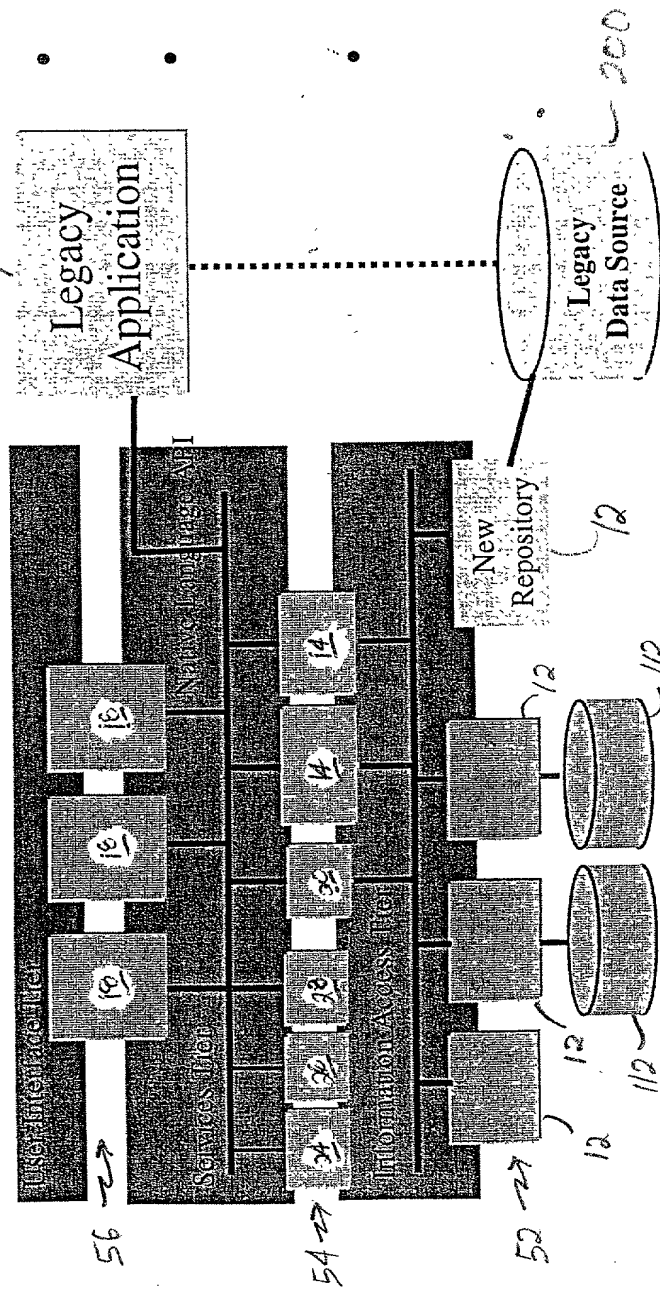
Integration with legacy systems

Full Integration
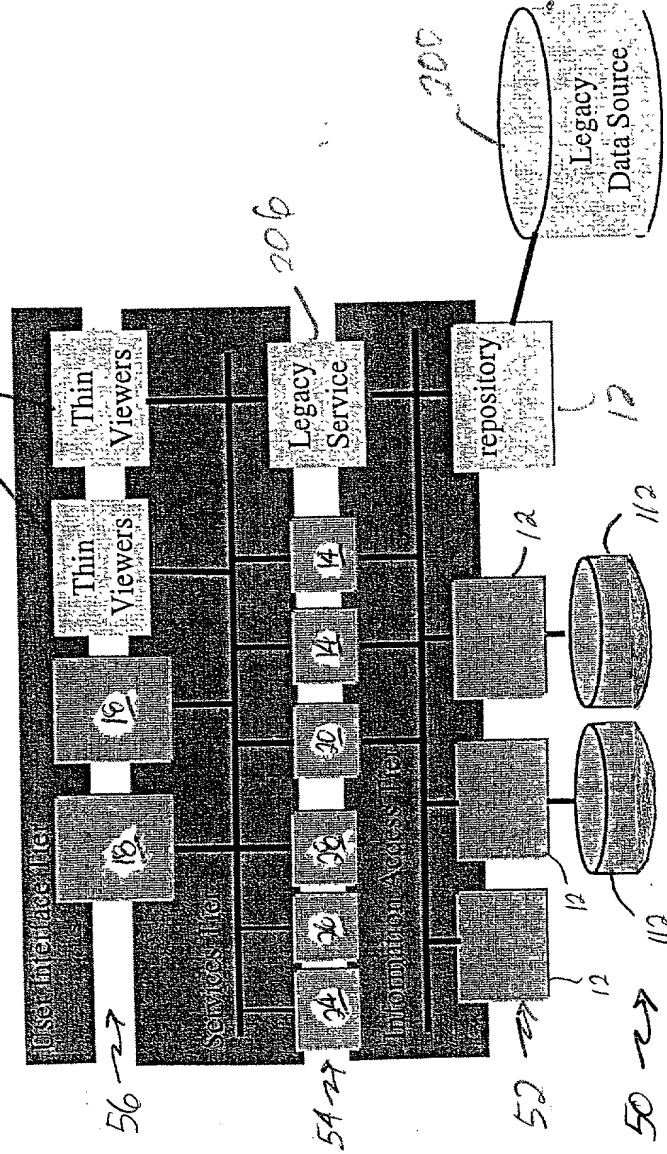
- Rewrite viewers in Java making them web-enabled and machine independent

- Legacy processing becomes a system component available for enterprise usage

- Lowers maintenance cost

- Duplicate functionality removed across the enterprise

- Each enhancement is available to the entire enterprise

Thin Viewers

Thin Viewers

Legacy Service

repository

Legacy Data Source

User interface Tier

Services Tier

Information Access Tier

SYSTEM   Infrastructure   Legacy System

Fig. 13

Importance of Data-Centric Collaboration Framework

- Framework is applicable to most domains
- Small tools extend overall capability
  - Build domain or analyst specific tools--not systems
  - Adding single collaborative capabilities results in exponential growth of overall system capability
- Collaboration integral to framework
  - Instead of pasting images onto a whiteboard, collaborate on the tool itself using whiteboarding layer
  - No special logic needed in tools to support collaboration
- Supports legacy applications
  - Data is shared and not replicated, so changes to the data by legacy tools propagate to collaborative tools.

Fig. 14

Collaboration Application Management

- Runs in a Web Browser or as a separate Unix or NT Application.
- Provides ease of access to room and conference information
- Allows multiple saved workspaces consisting of conferences and tools.
- Allows drag and drop of documents to viewers
- Allows easy navigation between conferences and rooms.

Fig. 15

Collaborative Application Management

- Multiple Active Rooms/Places that contain Documents, People, Tools and Conferences
- Within a Conference a person or group of people analyze information and interact to solve problems
- People within a Conference are shown
- Documents being used within a Conference are shown
- Tools that are connected to documents are shown
- User can minimize/maximize/remove individual tools, tools associated with a conference or tools associated with a room
- Documents are dragged and dropped onto a conference from the document server window

File Edit Views Tools Chat

- Air Intel. Rm - Korea
- SAM-AAA Intel. Rm - Korea
  - NV
  - VAT
  - OpsPicture
    - Documents
      - KoreanCities
        - MapViewer
        - PlotViewer
        - PropertiesTool
      - OperationsChat
        - ChatViewer
      - AirborneSignals
        - MapViewer
        - DataList
      - KoreanFacilities
        - DataList
        - PropertiesTool
      - wbCommsNetwork
        - MapViewer
      - ThreatSignals
        - MapViewer
      - TADILTracks
        - PlotViewer
        - MapViewer
    - People
      - doug
      - Bob
      - Shirley
  - Targeting Rm - Korea

Fig. 16

Dynamic Repository Query &
Document Management

- Dynamically learns about repository

- Gets attribute meta-data from repository

- Creates agent representing standing query

- Results become a document which may be used for collaboration

New Standing Query Document

Name: ThreatSignals
FeatureType: Signal...

- Signal Information
  - Signal Modulation Mode Information
    - Signal Modulation Type
  - RF
    - Carrier Frequencies
    - Sequence of RFs
    - Carrier Frequency Stats
    - Fundamental Frequency Source
    - Carrier RF Type
  - Pulse Modulation Mode Information

Hide Filter   OK   Cancel

Carrier RF Type Di...

Pulse Constant RF
Pulse Simultaneous RFs
Pulse Sequencial RFs
Pulse to Pulse Random RF
Pulse to Pulse Periodic RF
Pulse to Pulse Adaptive RF
Burst to Burst Agile RF

OK   Cancel

New Standing Query Document

Name: Cities
FeatureType: cities ...

Row
- CITYCODE
- PLACECODE
- NAME
- STATE
- COUNTRY range[Korea : Korea]
- LATITUDE
- LONGITUDE
- PC
- LC

COUNTRY Dialog
Min: Korea   Max: Korea
OK   Cancel

Folder Browser

A
ThreatSignals
KoreanCities
KoreanInstallations
AirborneSignals
TADILTracks
wbCommsNet

B

doug's Personal Folder
rich's Personal Folder
Scott's Personal Folder
Anne's Personal Folder
Trebor's Personal Folder
Shirley's Personal Folder
Bob's Personal Folder
Jeff's Personal Folder
Kurt's Personal Folder
Targeting Rm - Korea Contex
SAM-AAA Intel. Rm - Korea C
Air Intel. Rm - Korea Context

Fig. 17

- The BBN Open Map Viewer was selected since it supported layering and a standards-based interface. No license fee is required. It is an Open Source component.

- A Mercator projection is shown with items colored via the data model

- A configurable pop-up menu can be seen

- Integration with commercial and legacy map products is based on OpenGIS standard APIs.
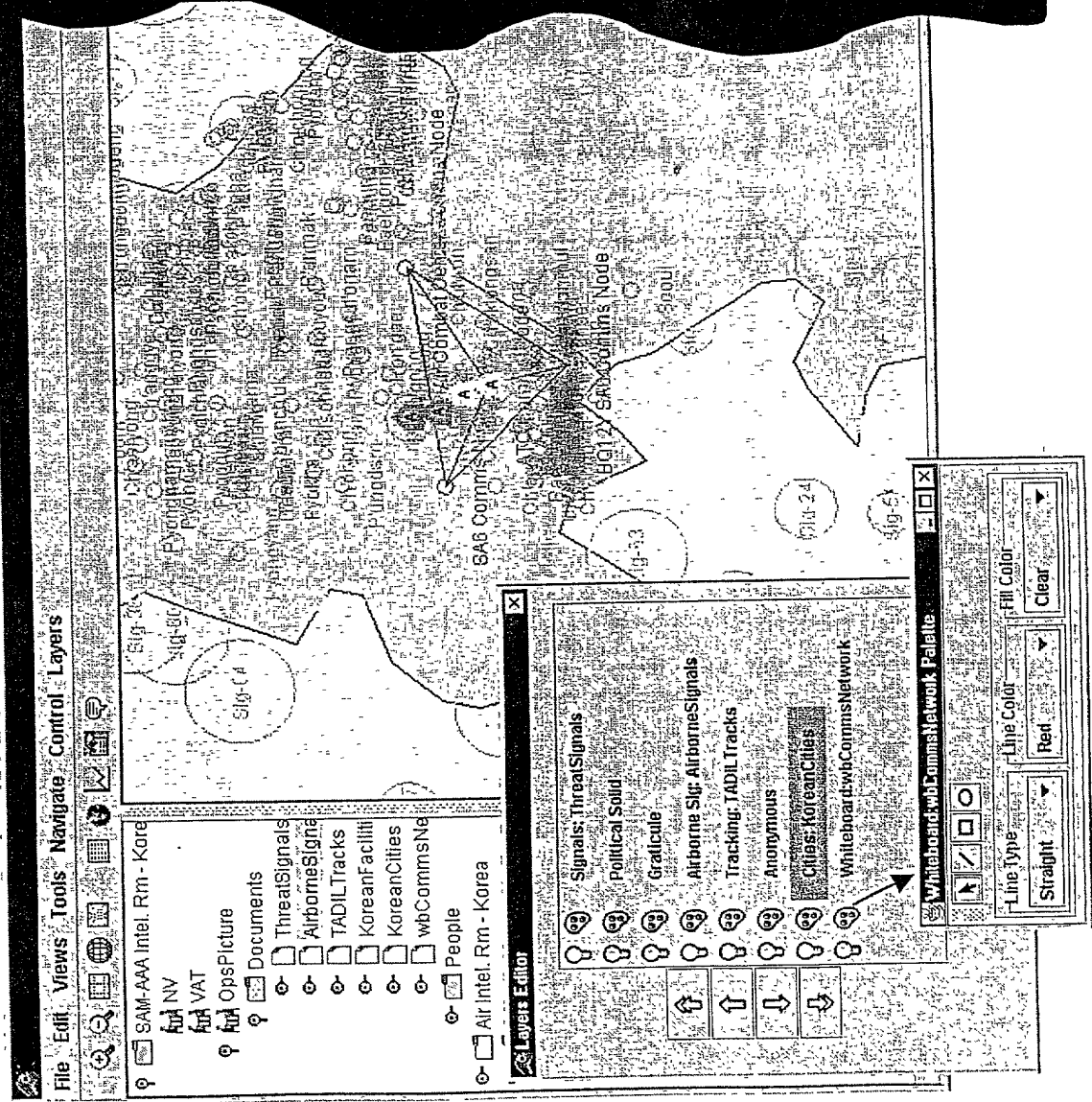
Fig. 18

Extended Properties Editor

- Dynamically learns information schema from repository

- Attaches extended extended properties to data in the data channel

- Applied rules run as agents within the channel

- Extended Properties
  - Color
  - Highlight
  - Visibility
  - Label
  - Symbol
  - ......

Fig. 17
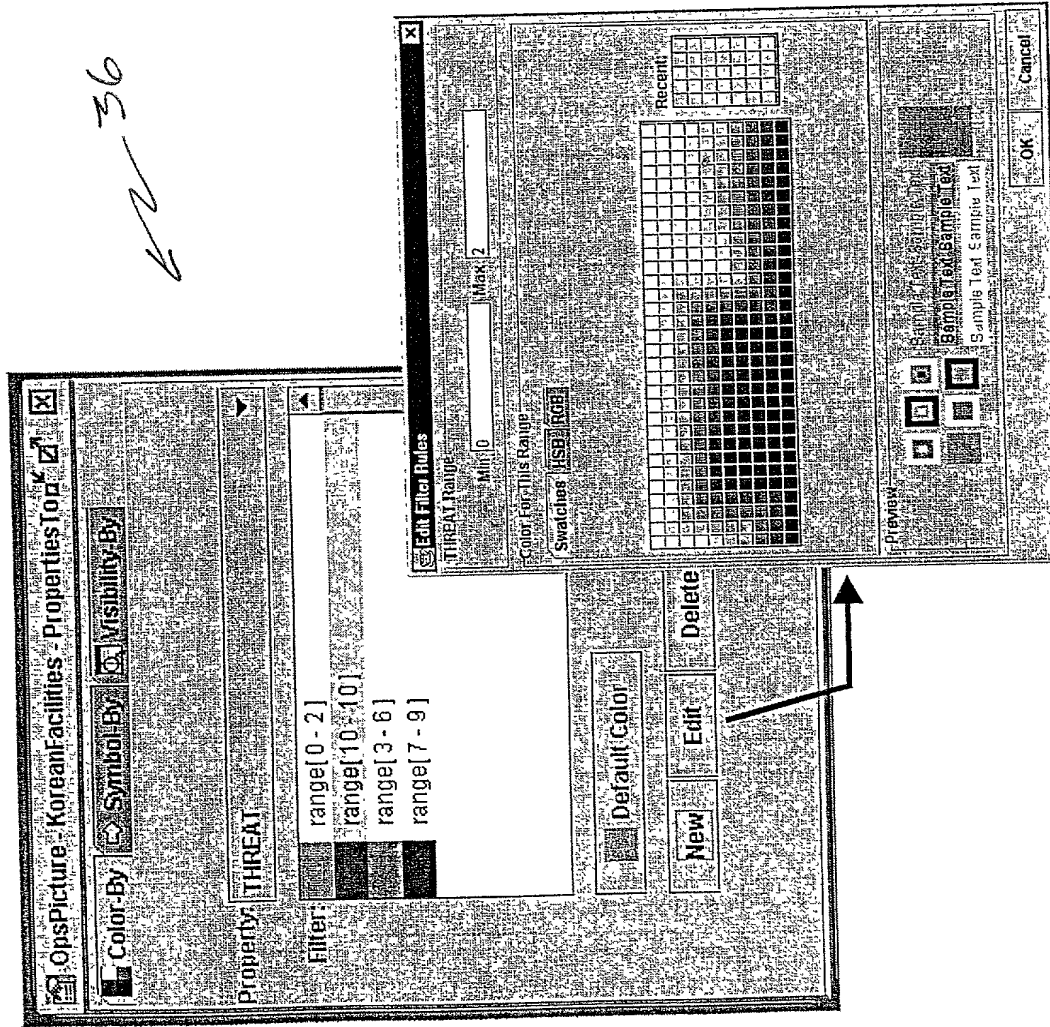
X-Y Plotter

- Select X and Y Attributes From List provided by Repository
- Re-order displays
- Zoom/Pan in any display independently or dependently

OpsPicture - ThreatSignals - PlotViewer

Signal Number

Signal Number

Signal Number

CRYSTAL

3 plot(s) selected

Plot Axis Selection

Source

Carrier Frequencies
Sequence of RFs
Carrier Frequency Stats
Fundamental Frequency Source
PRI Values
PRI Sequence
PRF
PRI Stats

Destination

Carrier Frequency Stats
PRI Stats
Crystal

X-Axis Selection:

Signal Number

OK   Apply   Cancel   Help

Fig. 20

List Viewer
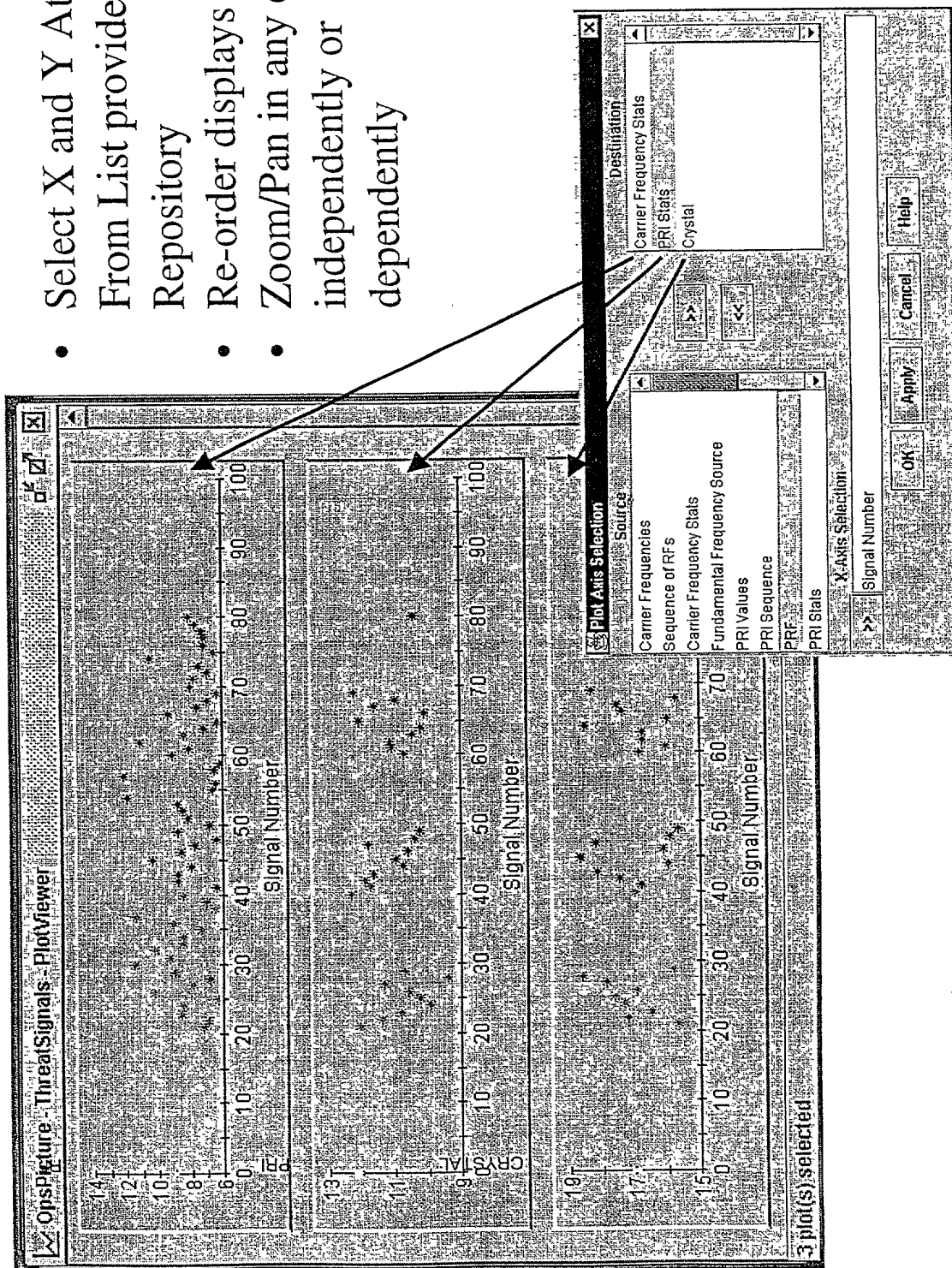
- Sorting
- Row Selection
- Row Coloring
- Row Hiding
- Choose Attributes to View

Fig. 21

Chat Tool

18

OpsPicture - OperationsChat - ChatViewer

```
Doug: TADIL is reporting an inbound plane
Doug: Does anyone know what type of planes those are
bob: I got an ELINT hit against track #52.  It looks like it is a MIG Fighter
Aircraft
shirley: Based on the comms between the two aircraft they are planning to cross
the boarder and pull our fighters into a trap
shirley: They have a network of SAM's ready to take out our pursuing fighters
shirley: Have we seen any fighter activity?
bob: I just got an SA-6 TT hit on the SUNAN site.  We had better let the AWACS
know, and amplify those TADIL tracks with our ID informaiton
```

- Chat supports multi-user conversations from multiple conferences in multiple contexts

- People connect to a document and communicate

- People in the same conference see the same visualization properties like color and visibility of participants inputs

- Conversations are persistent over time
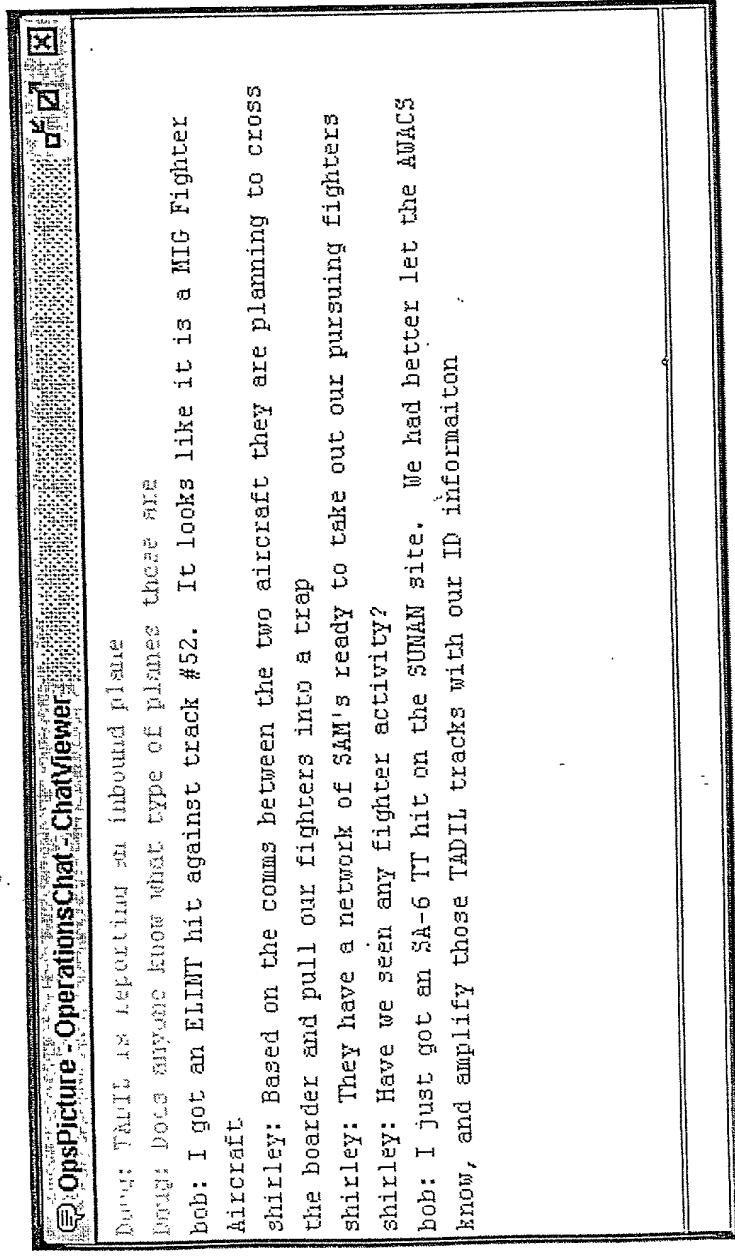
Fig. 22

Performance Metrics
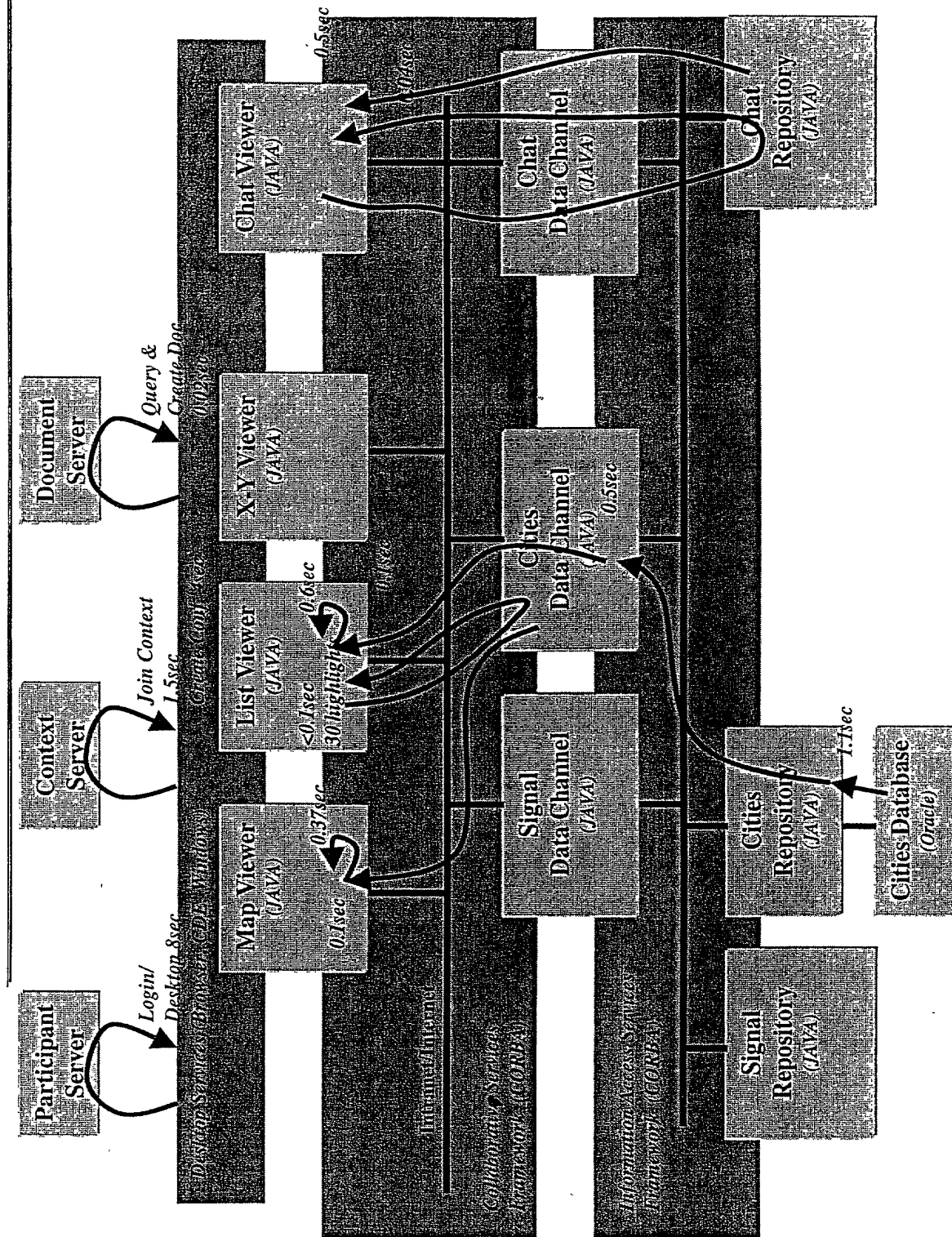
Fig. 23

USIGS - Geospatial and Imagery Access Services Specification

USIGS
Common Object Specification
(UCOS)

Create Request

Submit Query Request

Submit Standing Query Request

Library

Manager

Request

Products

GIAS Client

1) get_manager_types

2) get_manager

3) sent_request

4) register_callback

5) callback_notify

Callback

Polling

Blocking
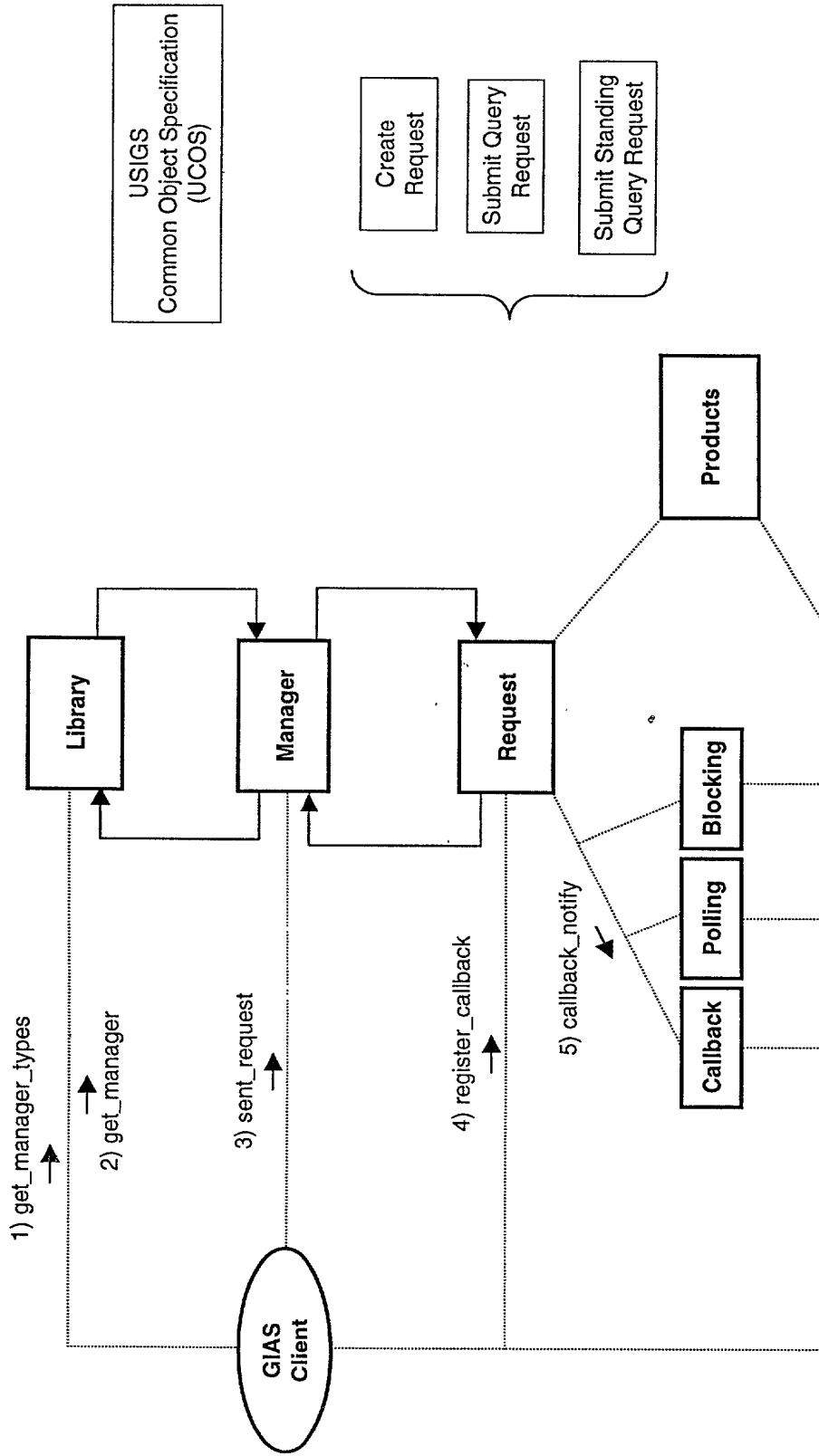
- Dynamic discovery of information sources
- Dynamic discovery of access techniques
- Synchronous, Asynchronous, Polling Access mechanisms
- Clients autonomous request executing within the data environment
- All Interfaces and Structures represented within IDL (UCOS - DAG)

Fig. 24

Information Access Services

Reporter

20

Library Services

Library Services

16

Knowledge Repository Services

12

1) get query mgr

2) submit query

Client

Library

Feature Types Mgr

3) create

8) get DAG

Property Def

Feature Type

Standing Query Request

4) create

Feature Collection

7) create feature per domain object

5) execute query

Installation Definition

Repository

6) execute query

Data Store

OODBMS, RDBMS, File

Cities Data Source

Feature

Property

In this example the Client is accessing the query result directly without using the Collaborative and extended properties of the Data Channel
- Client dynamically learns about Repository Features via Library Services
- Client submits a query via the Library Services
- Client retrieves information resulting from the query via the returned Request and Feature Collection

Fig. 25

Library Virtual Access
Requesting Information

- Client knows only about Library
- Client learns about Feature Types through Feature Type Mgr
- Client submits query through the Standing Query Mgr
- Repository and Feature Collection adapt to Database Particulars

Client

Library

StandingQueryMgr

FeatureTypesMgr

QueryRequest
(Feature Collection)

InstallationType
(Feature Type)

InstallationRepository
(Repository)

InstallationFeatureCollection
(Feature Collection)

FacilityType
(Feature Type)

QueryRequest
(Feature Collection)

FacilityRepository
(Repository)

FacilityFeatureCollection
(Feature Collection)

UnitType
(Feature Type)

UnitRepository
(Repository)

QueryRequest
(Feature Collection)

UnitFeatureCollection
(Feature Collection)

Fig. 26

Library Virtual Access
Accessing Current Information

FeatureTypesMgr

Library

StandingQueryMgr

Client

ClientCallback

UnitType
(Feature Type)

UnitRepository
(Repository)

QueryRequest
(Feature Collection)

UnitFeatureCollection
(Feature Collection)
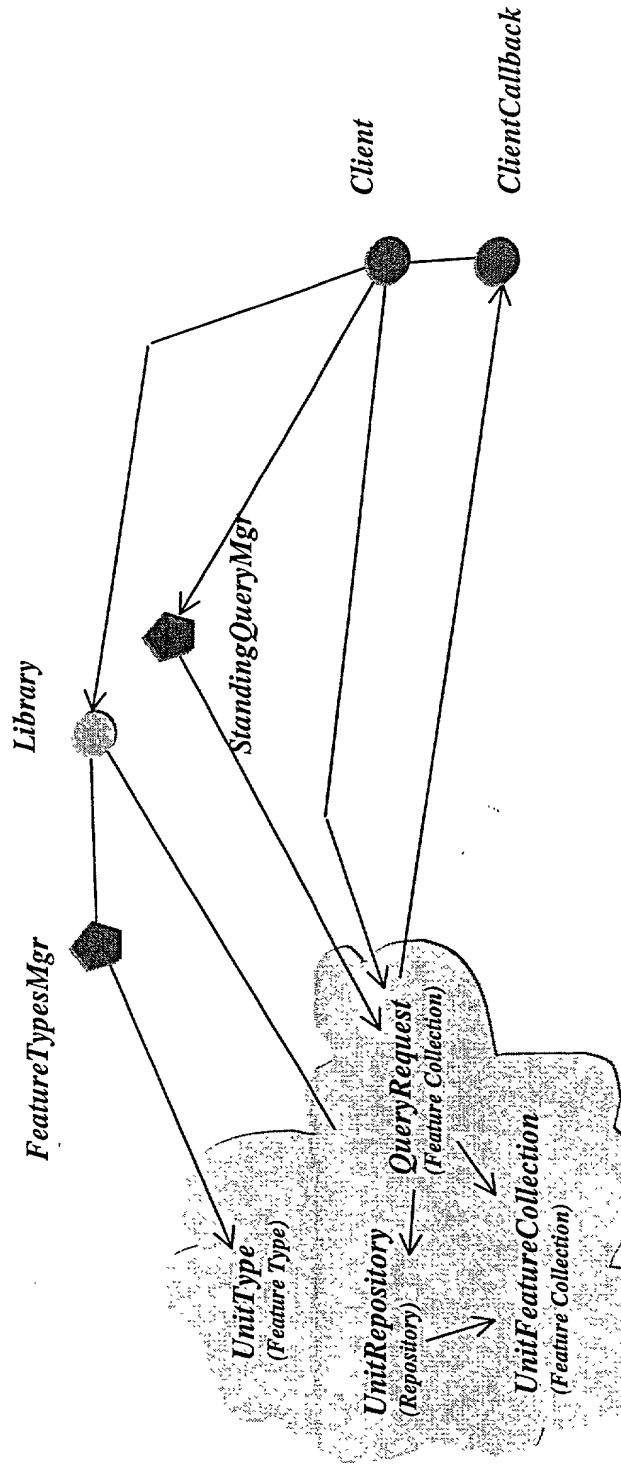
- Client Learns about Standing Query Mgr through Library
- Query Manager returns a reference to a Request Object for each client query method invocation
- Client interacts with Request for Query Control and Status
- Request supports Synchronous, Polling, and A-Synchronous Client interfaces. ClientCallback is used for A-Synchronous feedback on query state

Fig. 27

```
                    ┌──────────────────────────────────────┐
                    │           <<Interface>>              │
                    │  SubmitStandingQueryRequestFactory   │
                    ├──────────────────────────────────────┤
                    │ ▪create()                            │
                    └──────────────────────────────────────┘
                                     │
                                     ▽
        ┌─────────────────────┐         ┌─────────────────────────┐
        │    <<Interface>>    │         │      <<Interface>>      │
        │    RequestFactory   │         │   CreationRequestFactory│
        ├─────────────────────┤◁────────├─────────────────────────┤
        │ ▪get_type()         │         │ ▪create()               │
        └─────────────────────┘         └─────────────────────────┘
                   △
                   │
        ┌──────────────────────────┐
        │      <<Interface>>       │
        │  SubmitAgentRequestFactory│
        ├──────────────────────────┤
        │ ▪create()                │
        └──────────────────────────┘
```

Fig. 28

<<Interface>>
FeatureTypesMgrComponent
initLibrary()

<<Interface>>
FeatureTypesMgr
get_feature_types()
get_feature_type()
get_data_source_names()

<<Interface>>
CreationRequestMgrComponent
initLibrary()

<<Interface>>
RequestManagerComponent

<<Interface>>
CreationRequestMgr
create_request()

<<Interface>>
LibraryManagerComponent

<<Interface>>
StandingQueryMgrComponent
initLibrary()

<<Interface>>
LibraryManager
get_library()
get_property_names()
get_properties_values()

<<Interface>>
RequestManager
get_active_requests()
delete_request()
get_default_timeout()
set_default_timeout()
get_timeout()
set_timeout()

<<Interface>>
StandingQueryMgr
submit_standing_query()

<<Interface>>
AgentMgr
submit_agent()

<<Interface>>
AgentMgrComponent
initLibrary()

Fig. 29

```
                ┌─────────────────────────────────┐
                │          <<Interface>>          │
                │  RepositoryBasedLibraryComponent│
                │                                 │
                └─────────────────────────────────┘
                                △
                                │
    ┌──────────────────────────────────────────────────────┐
    │                    <<Interface>>                      │
    │                RepositoryBasedLibrary                 │
    ├──────────────────────────────────────────────────────┤
    │  ▓get_repositories()                                  │
    │  ▓get_repositories_by_feature_name()                 │
    │  ▓get_repositories_by_data_source_name()             │
    │  ▓get_repository_by_feature_and_data_source_name()   │
    │  ▓add_repository()                                    │
    └──────────────────────────────────────────────────────┘
                                △
                                │
                ┌───────────────────────────────┐
                │          <<Interface>>        │
                │             Library           │
                ├───────────────────────────────┤
                │  ▓get_manager_types()         │
                │  ▓get_manager()               │
                │  ▓get_library_description()   │
                │  ▓add_manager()               │
                └───────────────────────────────┘
                                △
                                │
                ┌───────────────────────────────┐
                │          <<Interface>>        │
                │        LibraryComponent       │
                │                               │
                └───────────────────────────────┘
```

Fig. 30

<<Interface>>
CreationRequest

<<Interface>>
Request

- get_request_description()
- get_status()
- cancel()
- register_callback()
- free_callback()
- get_request_manager()

<<Interface>>
CollectionResultRequest

- get_results()

<<Interface>>
CreationRequestComponent

- init()

<<Interface>>
RequestComponent

- init()

<<Interface>>
SubmitAgentRequest

- complete()
- set_number_of_hits()

<<Interface>>
SubmitStandingQueryRequest

- complete()
- pause()
- restart()
- set_number_of_hits()

<<Interface>>
SubmitAgentRequestComponent

- init()

<<Interface>>
SubmitStandingQueryRequestComponent

- init()

Fig. 31

```
              <<Interface>>                              <<Interface>>
                 Feature                                 FeatureType

  property_exists()                          get_parents()
  get_property()                             get_children()
  set_property()          +feature_type      property_def_exists()
  delete_property()       ─────────────▶     get_property_def()
  get_property_sequence()                    get_property_def_sequence()
  get_dag()                                  get_dag()
  dispose()                                  dispose()



              <<Interface>>
             FeatureCollection

  add_element()
  merge()
  insert_element_at()
  replace_element_at()
  remove_element_at()
  create_iterator()                      <<Interface>>
  get_collection_feature_type()     FeatureCollectionComponent
  announceChange()              ◁──
  getFeaturesAndDAGs()                 initFeatureType()
  getSpecificFeatureAndDAG()
  getSpecificFeaturesAndDAGs()
  getSpecificFeature()
  getSpecificFeatures()
  create_feature_ids()
  remove_element_by()
  remove_elements_by()
  remove_all()



              <<Interface>>                              <<Interface>>
             FeatureIterator                            FeatureRepository

  next()                                     get_request_factory()
  next_n()                                   get_request_factory_types()
  advance()                                  get_feature_type()
  current()                                  get_feature_type_name()
  get_property()                             get_data_source_name()
  get_properties_sequence()                  execute_query()
  reset()
  more()
  dispose()
```

Fig. 32

Data Channel Services Framework

Fig. 33

Versioning Data Changes in the Data Channel

Desktop Client

IARManager

1) Start viewer (DataChannel)

Viewer

→

View

Maintains the most recently received version

ItemPresentation

Model Change events

2) Get DataModel
3) Register for events

4) Get DAG Definition
5) GetSelectedModelChanges(-1)
6) GetSelectedModelChanges(version)

- Users select documents to be processed by tools
- The Desktop Manager provides the tool with a handle to the Data Channel
- The tool learns about the data by asking the channel for the the metadata about the collection that the channel maintains
- The tools registers to receive any updates that the channel receives

- The Tool asks the channel for only the information needed for rendering its display
- As events are received, the tool asks for any updates since the last version of the model that was received.
- Step 6 requests all changes since the indicated version

Conference Manager

Data Channel

DataCentricConference

∨ m

DataCentricChannel

events

DataModel

∨ m

Item

Maintains DAG of Properties per Item

∨ m

Property

Maintains the current version for each item

Fig. 34

OpenGIS Simple Features Specification
Understanding a Feature Collection

```
Feature
Collection
    |
    | groups
    v
  Feature  <------------ defines --- Feature
    |                                 Type
    | * Manages                        |
    v                                  | groups
  DAG                                  | *
    |                                  v
    |                                DAG
    |                                Def
    |                                  |
 +--+-------+                     +----+------+
 |  |       |                     |    |      |
 v  v       v                     v    v      v
Nodes Edges Properties          Nodes Edges Property
                                              Def
```

Describes the actual structure and values of each domain object

Describes the possible structure and attributes of a domain object

Fig. 35

Directed A-Cyclic Graph (DAG)

Aircraft Node

Equip. List Node

Al Radar Node

type property

Comms Node

type property

Aircraft Pilot Node

name property

call sign property

Edge

WKSNode
(from simplefeatures)

first : int
last : int
WKSNode()
WKSNode()
toString()

WKSEdge
(from simplefeatures)

source : int
destination : int
WKSEdge()
WKSEdge()
toString()

+nodes[]

+edges[]

WKSFeatureAndDAG
(from simplefeatures)

featureid[] : byte
WKSFeatureAndDAG()
WKSFeatureAndDAG()
toString()

+properties[]

WKSNVPair
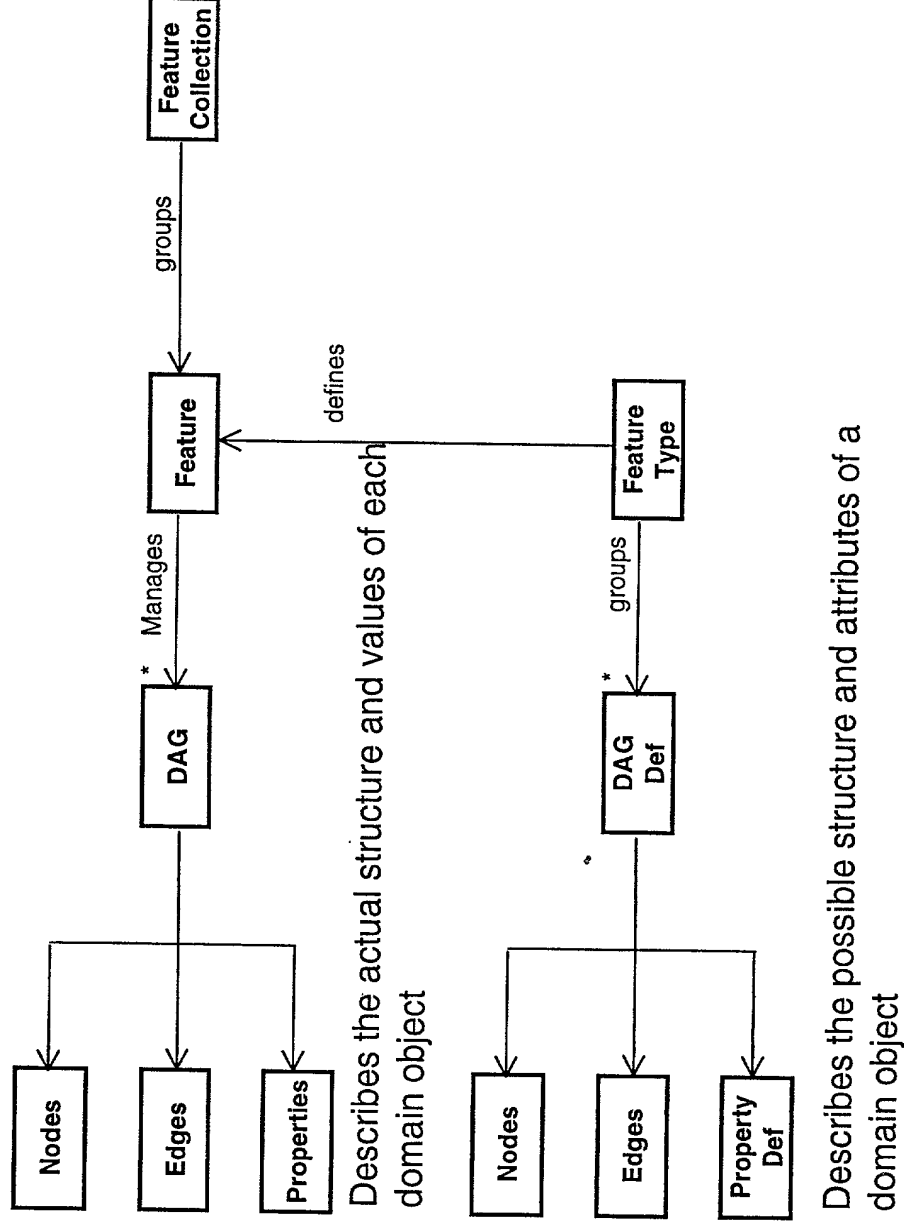*(from simplefeatures)*

WKSNVPair()
WKSNVPair()
toString()

Fig. 36

Directed A-Cyclic Graph (DAG)

node #1 = name:"Airplane"; type:"element"

property #1 = "name": (SimpleFeatures.WKSString)"Airframe 321"
property #2 = "type": (SimpleFeatures.WKSString)"Mig29"

edge #2
source node = #1
destination node = #3

edge #4
source node = #3
destination node = #5

node #3 = name:"EquipmentList"; type:"nodelist"

edge #1
source node = #1
destination node = #2

edge #3
source node = #3
destination node = #4

node #2 = name: "Pilot"; type:"element"

property #4 = "start time": (SimpleFeatures.WKSAbsTime) 12/1/99 12:32:12.1222
property #5 = "stop time": (SimpleFeatures.WKSAbsTime) 12/1/99 12:32:23.233
property #6 = "name": (SimpleFeatures.WKSString) "Joe Smith"
property #7 = "all sign": (SimpleFeatures.WKSString) "Idaho"

node #4 = name: "AI Radar"; type:"element"

property #11 = "rf range": (SimpleFeatures.WKSDoubleRange) 10.0 - 14.0
property #12 = " bandwidth":(SimpleFeatures.WKSDouble) 1.0
property #13 = "pw range":(SimpleFeatures.WKSDoubleRange) 0.1-100.0
property #14 = "polarization type";:(SimpleFeatures.WKSCategory) "all"
property #15 = "sensitivity":(SimpleFeatures.WKSCategory) "sidelobe"
property #16 = "modulation on pulse":(SimpleFeatures:WKSCategory) "IMOP", "UMOP"
property #17 = "dwell duration": (SimpleFeatures.WKSTime) 00:00:00.1000
property #18 = "data type": (SimpleFeatures.WKSCategory) "pdw"
property #19 = "aircraft side": (SimpleFeatures.WKSCategory) "left"
property #20 = "location":(SimpleFeatures.WKSBoolean) yes

IDL DAG Structure Passed

node #1
node #2
node #3
node #4
node #5

edge #1
edge #2
edge #3
edge #4

prop #1
prop #2
prop #3
prop #4
prop...
prop #10
prop #11
prop...
prop #20

Fig. 37